

II. CLAIM AMENDMENTS

1. (Previously presented) A method of preparing object files on a first device that are used for rendering two-dimensional images of three-dimensional objects on a second device by processing object files containing data defining triangle primitives representing at least one three-dimensional object to determine a rendering order in which said triangle primitives may be rendered for creating a two-dimensional image in which the hidden surfaces are removed comprising the steps of:

- a) storing a plurality of planar triangle primitives corresponding to a plurality of surface portions of said at least one object in an object file, wherein said plurality of triangle primitives are represented by vertex data arranged according to an original order,
- b) selecting a triangle that does not potentially occlude any of the unprocessed triangles to start a triangle strip,
- c) selecting a neighboring triangle to add to the last triangle in the triangle strip,
- d) determining whether the selected neighboring triangle has the potential to occlude any of the unprocessed triangles,
- e) adding the selected neighboring triangle to the strip and going back to step c) if the selected neighboring triangle does not have the potential to occlude any of the unprocessed triangles,

f) going back to step b) to start a new triangle strip if the selected neighboring triangle has the potential to occlude any of the unprocessed triangles, and

g) repeating steps b) to f) until all the triangles in the object file are processed.

2. (Original) A method according to claim 1, in which step b) comprises a heuristic search to identify those triangles for starting a strip that potentially allow long strips to be formed, preferably by selecting triangles that do not have many unprocessed neighbors.

3. (Previously presented) A method according to claim 1, further comprising the step of dividing the object file into two, if at step b) it is impossible to find a triangle not occluding other ones of the unprocessed triangles.

4. (Previously presented) A method according to claim 3, in which the object is divided by using binary space partitioning techniques, preferably using any partitioning plane, so that at least one of the unprocessed triangles remains on a different side than the unprocessed triangles it has the potential to occlude, in which a plane of an arbitrary unprocessed triangle is used and the triangle is associated on the outward side of the splitting plane, thus leaving all the triangles that it may cover on the other side.

5. (Previously presented) A method according to claim 1, further comprising the step of determining the front side and the triangle planes of the triangles in the object file

wherein the determination if a selected triangle can occlude any of the unprocessed triangles comprises the steps of determining whether:

g) the selected triangle has one or more vertices on the front side of the plane of any of the unprocessed triangles, and

h) whether any of the unprocessed triangles positively identified in step g) have one or more vertices on the backside of the plane of the selected triangle.

6. (Previously presented) A method according to claim 1, comprising further the following steps before step b)

i) determining if the object has symmetry in the XY-plane of the object space,

j) storing a determined plane of symmetry in the object file, and

k) removing all vertices in the object file from one side of a determined plane of symmetry, and

repeating steps i) to k) for the XZ-plane of the object space and for the YZ-plane of the object space.

7. (Previously presented) A method according to claim 1, further comprising the step of:

l) transferring said object file or files from said first device to a second device after step f) or k) and storing the object files on said second device.

8. (Previously presented) A method according to claim 1, in which said first device comprises a higher performance image processing means and said second device comprises a lower performance image processing means.

9. (Previously presented) A method for displaying object files obtained by the method of claim 1, on said second device, comprising the steps of:

m) determining the viewing direction, and

n) determining the object order, preferably using the painter's algorithm and displaying said object or objects on said second device.

10. (Original) A method according to claim 9, further comprising the step of:

o) culling all triangles facing backwards relative to said viewing direction.

11. (Previously presented) A method according to claim 9, further comprising the steps of:

p) retrieving planes of symmetry stored in the object file, and

q) creating a duplication of the vertex data present in the object file, said duplication of the vertex data having the same order as the original vertex data and being stored in the object file after the original vertex data, and flipping the sign of the vertex values on the axis orthogonal to the respective retrieved plane of symmetry of the duplicated vertices.

12. (Previously presented) A method according to claim 9, further comprising the steps of

r) displaying the triangle strips in the same order as they were created, and

s) displaying the triangles in the triangle strips in the same order as they were added to respective ones of the triangle strips.

13. (Currently amended) A method according to claim 9, in which said second device comprises a lower performance processing means ~~such as used in a mobile communication terminal or a PDA.~~

14. (Previously presented) A method, according to claim 9, in which said second device is a mobile phone.

15. (Previously presented) A method according to claim 13, in which the image rendering process is carried out by the main CPU of the device, and all the further image rendering means is software embedded.

16-26. (Cancelled)